University of Saskatchewan
Department of Computer Science

CMPT 214 - Programming Principles and Practice

FINAL EXAM
June 29, 2004

160 Marks
Time: 180 Minutes
(OPEN BOOK. NO CALCULATORS)

Student's Name: _____
(Please print in BLOCK letters, surname first.)

Student Number: _____

Instructor's Name: David J. Callele and Dwight J. Makaroff

Special Instructions: The weight of each question is given in parentheses. The total number of marks is 160. The total time is 180 minutes. Budget your time.
This examination consists of 11 pages, including this cover page. **Check to ensure that this exam is complete.**

1. READ AND OBSERVE THE FOLLOWING RULES:
   - Answer each of the following questions in the space provided in this exam booklet. If you must continue an answer (e.g.in the extra space on the last page, or on the back side of a page), make sure you clearly indicate that you have done so and where to find the continuation. If you should find it necessary to do so, you have probably written too much.
   - Make all written answers legible; no marks can be given for answers which cannot be decrypted. Where a discourse or discussion is called for, be concise and precise.
   - If you find it necessary to make any assumptions to answer a question, state the assumption with your answer.
   - ALWAYS SHOW ALL YOUR WORK. A final answer, even if correct, without supporting work is not acceptable.

P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	TOTAL
10	25	15	12	20	10	22	10	18	18	160

1. (10 marks) Write a *bash* shell script called *validate* that takes, as a single command-line argument, a single file name. Each line of the input file is *expected* to have data in the format:

   ```
   name,grade
   ```

   Your shell script is to read the input file and ensure that each line in the input file is of the expected form. The *name* field can be composed of any characters that are not numbers in the range 0 to 9. The *grade* field can be composed of integers in the range 0 to 100 inclusive.

   The script *validate* must be constructed in a manner consistent with other utility scripts (and applications) that can be concatenated using the pipe (|) operator.

2. (20 marks) Given the following data set, give the regular expressions to match the defined patterns.

```
FirstModel.mdl
FirstModel_01.mdl
SecondModel_00.mdl
ThirdModel_01.mdl
FourthModel00.mdl
Fourthmodel_01.mdl
FourthModel_02.mdl
FifthModel_00.mdl
FifthModel_01.mdl
```

   - All strings that start with the letter F

   - All strings containing the sequence 01 or higher (e.g. 02, 03, etc.)

   - All strings that do not contain the string Model

   - All strings that do not contain a sequence number in the form _xx where x is a digit.

3. (5 marks) In the context of the bash shell, what is *command substitution*? Give an example that demonstrates how command substitution works and explain what your example does.

4. (5 marks) What is the difference between a *reference* in Java and a *pointer* in C?

5. (5 marks) Explain the relationship between *pointers* in C and the array construct [ ] in C. How are they the same? Different?

6. (5 marks) Explain how the *function pointer* and *struct* primitives in C can be used as building-blocks for implementing an object-oriented language.

7. (5 marks) Why is it beneficial to be involved in a software consortium? Choose 2 reasons and explain in your own words.

8. (3 marks) When using a Makefile, how does *make* determine which actions to carry out?

9. (4 marks) How are *patterns* in *awk* and *targets* in a *Makefile* the same? How are they different?

10. (20 marks) Write a utility program that implements the base functionality of *wc* (word count). Your program should take, as a single command-line argument, a single file name. Your program should write the number of newline characters (the escape sequence for the newline character is $\backslash n$), words and total number of characters (including non-printing characters) contained in the input file to the standard output. You may assume that either a space character or a period separates (delimits) the words and that no other character needs to be recognized as a separator (delimiter).

    You may use either bash, C, or Perl. Indicate which (it might not be obvious) language you choose, and why. Note: *You may not use* **wc** *in your answer :-), just in case you were tempted.*

11. (10 marks) Write a utility function *transpose* in C that transposes a two-dimensional array. The function *transpose* complies with the following interface declaration:

```
int* transpose(int* ipInputArray, int iRows, int iCols);
```

The function *transpose* must dynamically allocate memory for the transposed array result and return the address of the transposed array.

Array transposition is defined as follows:

Input Array: {{0, 1, 2},{{3, 4, 5}} Two rows of 3 columns

Output Array: {{0, 3}, {1, 4}, {2, 5}} Three rows of 2 columns

12. (6 marks) Write an *awk script* to find and print the maximum salary for employees with the string "John" in their name. Assume all salaries are positive real numbers.

    The format of the file is as follows:

    ```
    Name(string); Position(string); Salary (real)
    ```

13. (4 marks) Explain in your own words, the difference between

    #define DEBUG 1 and

    const int DEBUG = 1 and how this is handled by the compiler.

14. (6 marks) To improve the performance of a program, several approaches are possible. Explain in your own words the disadvantages of **two (2)** of these approaches.

15. (6 marks) Explain why Perl and the Web are good for each other.

16. (10 marks) Write a Perl program to filter a text file as follows:

    Find all occurrences of postal codes and change them to **H0H 0H0** for the Christmas mail season. Any line that does not have a postal code is to be printed as is to the standard output. Those lines with a postal code are to be written to *BOTH* the standard output and to a special log file (of your choosing) except for the following change: *The log file is only to have the original postal code, one postal code per line.* Assume that anything which matches the format of a postal code is a postal code.

17. (18 marks) The following code sample has (at least) 3 places where a run-time failure could occur. Identify **three (3)** errors (circle them in the source code). Suggest alternative code and justify your suggestion. Finally, suggest a test case for each error that would identify the error.

```
int scale(int* ipInput, int* iCols, int iScaleFactor)
{
  int x = 0;
  for (x = 0; x <= iCols; ++x)
  {
     ipInput[x] = *(ipInput + x) / iScaleFactor;
  }
  return(&ipInput);
}
```

18. (6 marks) When building a library, identify 3 considerations the implementor must make with respect to the user and why these considerations are important.

19. (6 marks) What are the dangers of starting to write detailed code too early in the project life cycle? What are the dangers of starting to write detailed code too late?

20. (6 marks) Describe briefly the reasons for choosing between clear box testing and black box testing.

THE END

The remainder of this sheet is for rough work.